Using Runtime Optimizations to improve Energy Efficiency in High Performance Computing

Sridutt Bhalachandra

Advisors: Dr. Allan K. Porterfield and Prof. Jan F. Prins
Mentors: Dr. Stephen L. Olivier and Dr. Robert J. Fowler

Energy efficiency in high performance computing (HPC) will be critical to limit operating costs and carbon footprints in future supercomputing centers. With both hardware and software factors affecting energy usage there exists a need for dynamic power regulation. This dissertation highlights an adaptive runtime framework that can allow processors capable of per-core specific power control to reduce power with little performance impact by dynamically adapting to workload characteristics. Monitoring of performance and power regulation is done transparently within MPI runtime and no code changes are required in the underlying application. In the presence of workload imbalance, the runtime reduces the frequency on cores not on the critical path thereby reducing power without deteriorating performance. This is shown to reduce run-to-run performance variation and improve performance in certain scenarios. For applications plagued by memory related issues, new memory metrics are identified that facilitate lowering power without adversely impacting performance.

1 Research and Importance

In recent times, the fundamental drive in supercomputing to increase peak performance by adding increasing number of power hungry components has waned, with the shift in focus to energy efficiency to mitigate large operating costs. The Exascale Computing study [KBB+08] has set a definitive power challenge to deliver Exascale performance using only 20 Megawatts (MW). Today's top supercomputers with less than $1/10^{th}$ exascale performance already consume in excess of 10MW. A commensurate increase in power is no longer feasible. Nominally homogeneous computing elements exhibit heterogeneous performance and limiting power increases the performance variation [RAdS+12, PFB+15]. Software variations due to varying data locality, compiler optimizations and number of threads amount to 20% in general with over 2x in extreme cases [POBP13]. These variations suggest a need for dynamic power regulation to achieve savings in energy.

Multi-core CPUs often have load imbalances between their cores that are either fixed or dynamic. Most HPC applications use parallel programming paradigms with barrier synchronizations that require all processors to finish before proceeding further. Power and thermal constraints affecting chips differently cause on-chip mechanisms that control operating frequency to also vary. This makes performance vary between sockets for even perfectly balanced parallel applications. HPC applications need access to memory often, and sometimes even I/O. These memory operations are seldom explicit, making it difficult for the

operating system to stall (or switch off) cores and reduce power while waiting on memory. For applications that heavily utilize the memory sub-system slowing processor speed or CPU throttling shows little impact on performance while decreasing power [WPCB15, POBP13].

I propose to dynamically identify and utilize scenarios exhibiting computational workload imbalance and/or are constrained by memory to improve energy-efficiency both on conventional and power-limited systems. The differentiation from prior work lies in employing adaptive methods at runtime, and power control levers that have not been readily applied to the above two scenarios. An adaptive runtime framework (library) transparent to application is to be developed that will allow processors to reduce power with little performance impact by dynamically adapting to workload characteristics. Different core-specific power controls could either be employed separately or combined to enhance effectiveness of the framework. For applications plagued by memory related issues, we identify new memory metrics that facilitate lowering of power without adversely impacting performance.

2 Highlights

2.1 Using Dynamic Duty Cycle Modulation for energy-efficiency in HPC

On Intel processors before Haswell, Dynamic Voltage and Frequency Scaling (DVFS) affects all cores of a multi-core processor. Slowing the critical path slows execution. DVFS-centric research has focused on finding situations where the slowdown is greatly outweighed by the energy savings. Intel also supports Dynamic Duty Cycle Modulation (DDCM) where the effective frequency of each core can be adjusted nearly instantaneously by only gating a fraction of the clock cycles to that core. I propose use of DDCM as an alternative to improve energy-efficiency, and performance in power-capped environments. An adaptive runtime DDCM policy is to be developed to reduce power in unbalanced MPI applications. This work has been completed and published as [BPP15].

On Sandybridge systems, the adaptive DDCM policy for MPI was run on synthetic benchmarks and mini-apps – *miniAMR* and *graph500* on single and 16-node configurations. DDCM saved up to 13.5% processor energy on one node and 20.8% (for *miniAMR* with slowdown of less than 1%) on 16 nodes. By applying a power cap, DDCM effectively shifts power consumption between cores and improves overall performance. Performance improvements of 6.0% and 5.6% on one and 16 nodes, respectively, were observed. Thus, saving energy in power-limited systems is also seen to improve performance.

The policy was then validated with production applications like *ADCIRC, WRF and LQCD* [PFB+15]. For *ADCIRC* on 16 nodes, energy savings of over 10% with only a 1-3% slowdown is obtained. With a power limit of 50W, one version of the policy executes 3% faster while saving 6% in energy, and a second version executes 1% faster while saving over 10% energy. The effectiveness of DDCM is also seen for OpenMP with savings of 21% in energy and improvement of energy-delay product (EDP) by 16% [WPCB15]. With encouraging results on both shared and non-shared memory as well as production applications, DDCM is seen to be a viable alternative to achieve energy efficiency in HPC.

2.2 An Adaptive Core-specific Runtime for Energy-Efficiency

With addition of core-specific voltage regulators in Intel Haswell, DVFS can now slow down only non-critical cores like DDCM. I propose an Adaptive Core-specific Runtime (ACR) that allows processors with core-specific power control to reduce power with little performance

impact by dynamically adapting core frequencies to workload characteristics. A policy to combine the benefit of larger power reduction with DVFS owing to reduction in both voltage and frequency, and the ability of DDCM to lower the frequency beyond the operating range of DVFS is also proposed. This work has been completed and published as [BPOP17].

This work highlights a generic policy that effectively utilizes core-specific power controls. Our previous work (Section 2.1) aimed only at showing the efficacy of DDCM as an alternative to socket-wide DVFS. However, the present work offers a context for comparing DDCM (with its simple per-core hardware implementation and fast switching capability) and DVFS (more complex and costly to implement per-core but with potential for greater savings), and for showing how and when they can be used together. A transparent adaptive runtime framework (library) is implemented that throttles frequencies of cores not on the critical path of an MPI application using either DDCM, per-core DVFS or both.

The frame work is validated using six mini-apps (*miniAMR, miniFE, CloverLeaf, HPCCG, AMG, miniGhost*), and a real world application, ParaDis. The evaluation shows an overall 20% improvement in energy efficiency with an average 1% increase in execution time on 32 nodes (1024 cores) using per-core DVFS. An improvement in energy efficiency of up to 39% is obtained with the real world application ParaDis through a combination of speedup (11%) and power reduction (31%). The average improvement in performance seen is a direct result of the reduction in run-to-run variation and running at turbo frequencies.

As Exascale deploys over-provisioned systems that use per core power-limits in day-to-day operations, energy optimizations will be more important. Runtimes such as ACR will either allow more work to be run at one time by using less power or allow single applications to be run faster by allowing a higher power cap on critical cores than non-critical.

2.3 Improving energy efficiency in memory constrained applications

HPC would have been much easier if all the data required could fit in the cache of a processor, but rarely is this true. For certain classes of applications that heavily utilize the memory sub-system, slowing the processor speed or related approaches like CPU throttling has shown little impact on performance, with some cases showing performance improvement [WPCB15, POBP13]. There exists a need for solutions that can dynamically identify such opportunities. I propose identification of new metrics to detect applications that are constrained by memory and building an adaptive runtime policy using the new metrics to reduce energy wastage. This work has been completed and published as [BPO$^+$17].

We present an experimental memory study on modern CPU architectures, Intel Sandybridge and Haswell, to identify opportunities to reduce CPU frequency. Since the Last Level Cache (LLC) is shared, each core has to create a request for a particular memory location that is not in its private cache into the Table of Requests (TOR). Using uncore performance monitoring hardware counters, we identify a metric, *TORo_core*, that captures all valid requests in TOR.This metric detects bandwidth saturation and increased latency in the memory system, and is used in a dynamic policy to modulate per-core power controls.

The policy is evaluated when applied at coarse and fine-grained levels on six MPI mini-applications. The best energy savings with the coarse and fine-grained application of the dynamic policy is 32.1% and 19.5% respectively with a 2% increase in execution time in both cases. On average, the fine-grained dynamic policy yields a 1% speedup while the coarse-grained dynamic policy yields a 3% slowdown. Energy savings through frequency reduction not only provide cost advantages, they also reduce resource contention and create additional thermal headroom for non-throttled cores that can lead to improved performance.

3 Major work and ideas proposed in the Thesis

The goal of this thesis is to improve energy efficiency of HPC applications at runtime by matching the workload characteristics to power consumption utilizing underlying machine power controls. More specifically, the focus is on using core-specific power controls available in recent processor architectures. Two opportunities - computation workload imbalance and waiting on memory are identified to apply dynamic power regulation.

- **Dynamic Duty Cycle Modulation in High Performance Computing**: DDCM is shown to be an alternative to save energy, and improve performance in power-capped environments. The fundamental weaknesses of socket-wide DVFS can be overcome with DDCM as it has a per-core control with lower overheads allowing fine-grained core-specific clock frequencies. With power limits, slowing non-critical cores in software increases the available thermal headroom to the critical core improving performance.
- **Adaptive Core-Specific Runtime for Energy Efficiency**: The proposed ACR allows processors with core-specific power control to reduce power with little performance impact by dynamically adapting core frequencies to workload characteristics. Such a runtime could help alleviate heterogeneous processor loads that many future exascale applications will likely have. The MPI framework (library) is transparent to application and allows use of multiple power levers (DDCM, per-core DVFS or both).
- **Memory-Metric Policy for Reducing Energy**: My proposed characterization of HPC applications identifies a new metric that conforms to the memory behavior exhibited by many HPC mini-apps. The metric from this characterization is seen to be useful to construct dynamic runtime polices improving energy efficiency.

References

[BPO+17] Sridutt Bhalachandra, Allan Porterfield, Stephen L. Olivier, Jan F. Prins, and Robert J. Fowler. Improving energy efficiency in memory-constrained applications using core-specific power control. In *Proceedings of the 5th International Workshop on Energy Efficient Supercomputing*. ACM, 2017.

[BPOP17] Sridutt Bhalachandra, Allan Porterfield, Stephen L Olivier, and Jan F Prins. An adaptive core-specific runtime for energy efficiency. In *Parallel and Distributed Processing Symposium (IPDPS), 2017 IEEE International*, pages 947–956. IEEE, 2017.

[BPP15] Sridutt Bhalachandra, Allan Porterfield, and Jan F Prins. Using dynamic duty cycle modulation to improve energy efficiency in high performance computing. In *Parallel and Distributed Processing Symposium Workshop (IPDPSW), 2015 IEEE International*, pages 911–918. IEEE, 2015.

[KBB+08] Peter Kogge, Keren Bergman, Shekhar Borkar, Dan Campbell, W Carson, William Dally, Monty Denneau, Paul Franzon, William Harrod, Kerry Hill, et al. Exascale computing study: Technology challenges in achieving exascale systems. 2008.

[PFB+15] Allan Porterfield, Rob Fowler, Sridutt Bhalachandra, Barry Rountree, Diptorup Deb, and Rob Lewis. Application runtime variability and power optimization for exascale computers. In *Proceedings of the 5th International Workshop on Runtime and Operating Systems for Supercomputers*, page 3. ACM, 2015.

[POBP13] Allan K Porterfield, Stephen L Olivier, Sridutt Bhalachandra, and Jan F Prins. Power measurement and concurrency throttling for energy reduction in openmp programs. In *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2013 IEEE 27th International*, pages 884–891. IEEE, 2013.

[RAdS+12] Barry Rountree, Dong H. Ahn, Bronis de Supinski, David K. Lowenthal, and Martin Schulz. Beyond DVFS: A first look at performance under a hardware-enforced power bound. In *HP-PAC 2012: Proc. of the 8th Workshop on High Performance, Power-Aware Computing*, May 2012.

[WPCB15] Wei Wang, Allan Porterfield, John Cavazos, and Sridutt Bhalachandra. Using per-loop cpu clock modulation for energy efficiency in openmp applications. In *Parallel Processing (ICPP), 2015 44th International Conference on*, pages 629–638. IEEE, 2015.